Banco de Dados com JAVA

Dicas de Desenvolvimento

Alexandre Wolf

1- Passo (Iniciando)

- Instalar o banco de dados;
    - Criar um Arquivo (Banco de Dados);
    - Criar tabelas;
    - Dar permissões de acesso a ambos;

Exemplo

create database *BancoDados*;

use *BancoDados*;

create table clientes(codigo integer not null primary key, nome varchar(80) not null,endereco varchar(80),bairro varchar(80), cidade varchar(80), email varchar(100), telefone varchar(20));

grant all privileges on *BancoDados* to *Juca* identified by '*123*';

flush privileges;

grant all privileges on *BancoDados*.* to *Juca* identified by '*123*';

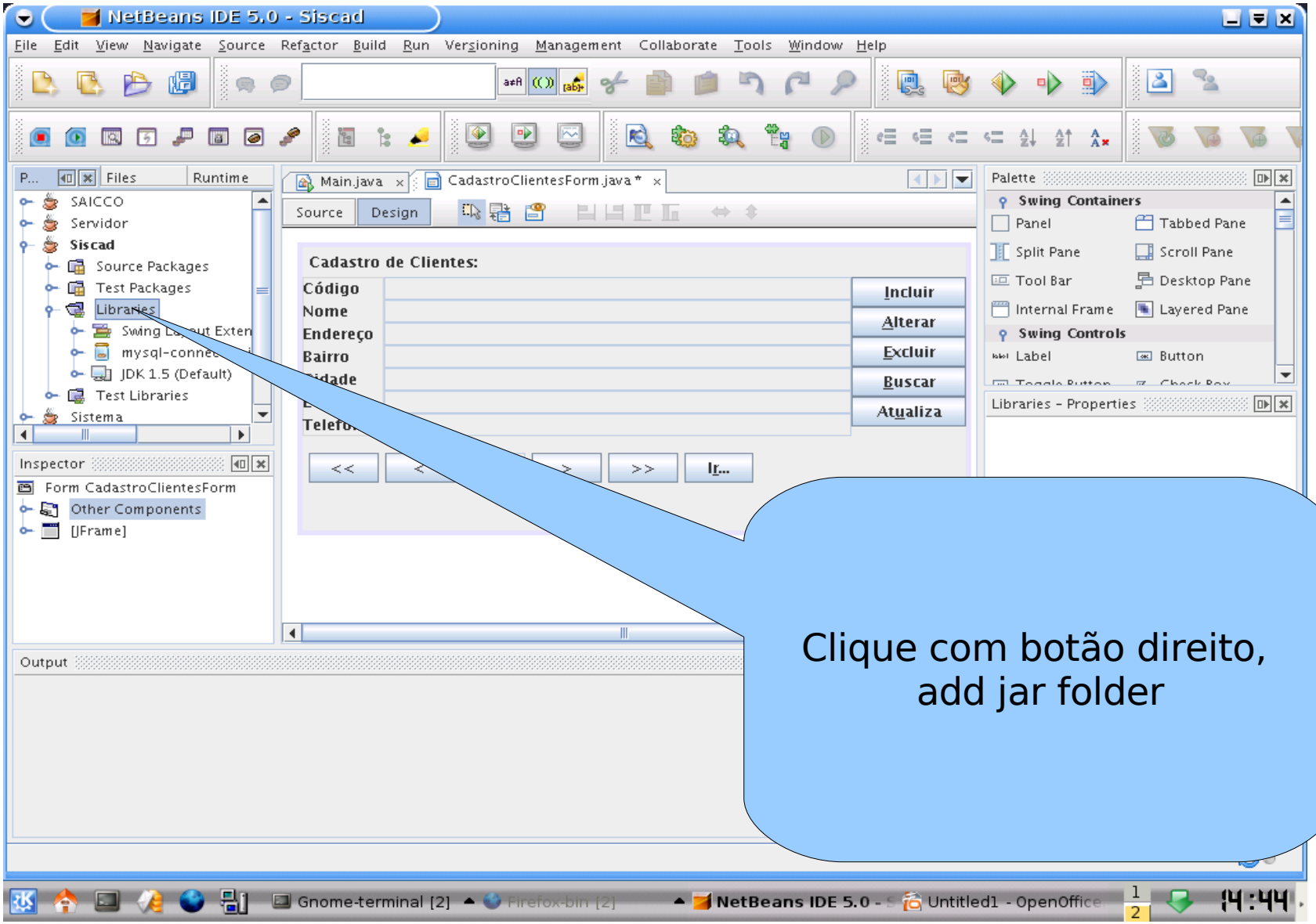
flush privileges;

2- Passo (Ambiente)

- Obter a versão mais estável de um *“Connector”* para o banco em questão;
  - Armazenar o mesmo no “classpath”, ou
  - .../<JavaInstallDir>/lib, ou
  - Qualquer diretório (Cuidado);

- No caso do NetBeans/Eclipse pode ser qualquer diretório, pois precisa importar a biblioteca (Classes);

Exemplo NetBeans

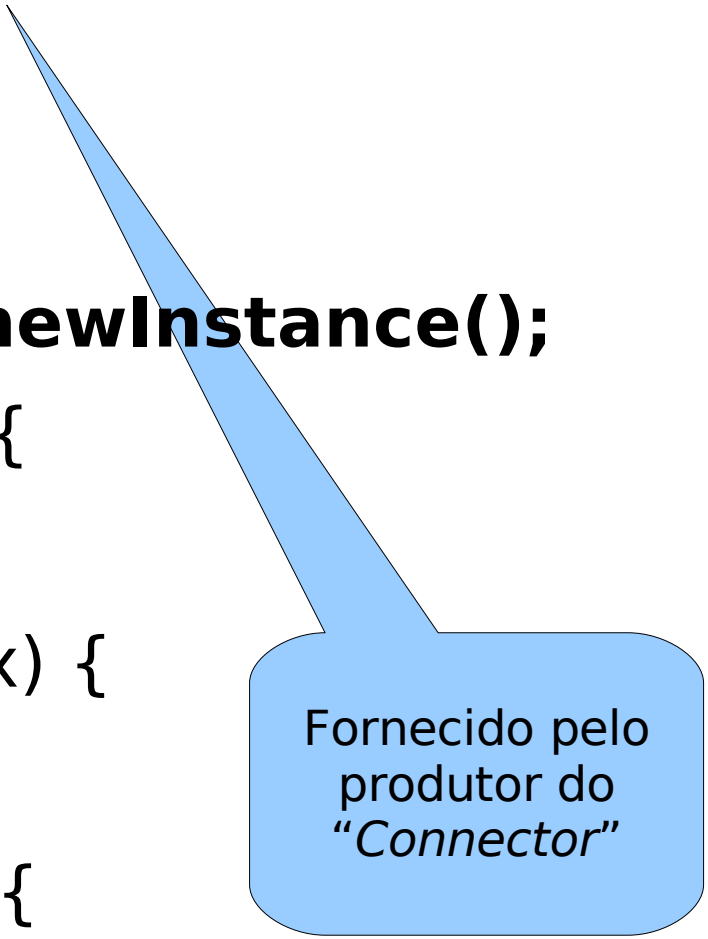


3 – Passo (Desenvolvendo)

- USAR A API: **java.sql.*;**
  - Registrar a classe de acesso (driver);
  - Estabelecer a conexão;
  - Executar procedimentos;

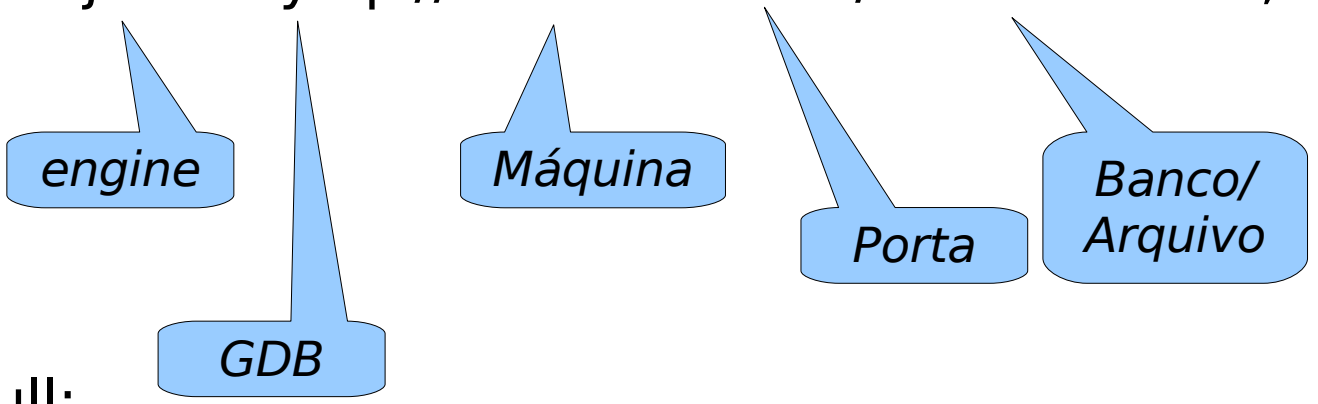
3.1 – Registro de Driver (Classe)

```
String NomeDriver = "com.mysql.jdbc.Driver";


try {
    Class.forName(NomeDriver).newInstance();
} catch (InstantiationException ex) {
    ex.printStackTrace();
} catch (ClassNotFoundException ex) {
    ex.printStackTrace();
} catch (IllegalAccessException ex) {
    ex.printStackTrace();
}
```



3.2 – Conexão com o Banco

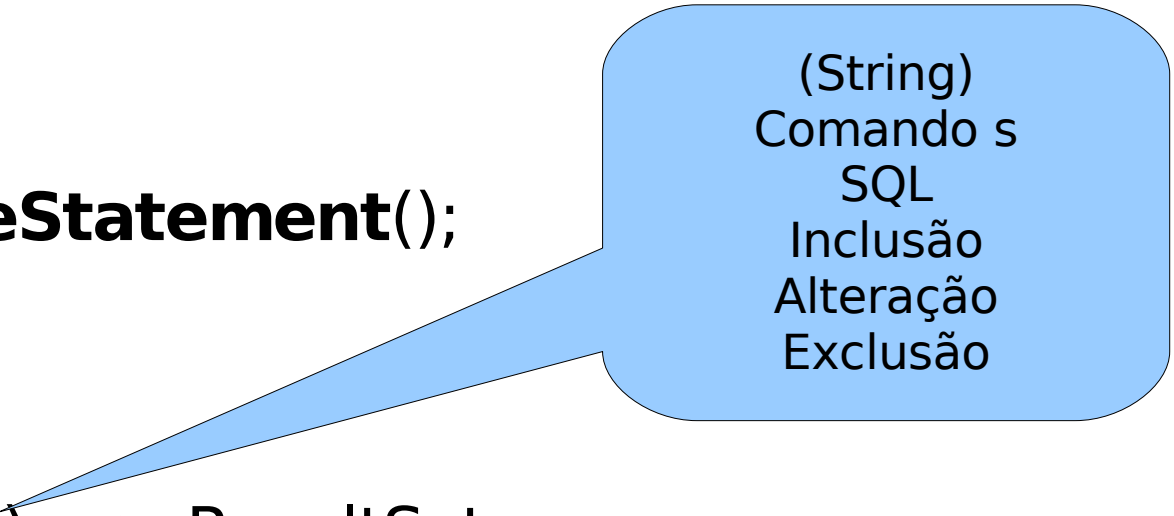
```
String LocalBancoDados = "jdbc:mysql://10.1.1.4:3306/BancoDados";
String Usuario = "Juca";
String Senha = "123";
```




```
Connection Conexao = null;
try {
    Conexao = DriverManager.getConnection(LocalBancoDados,
                        Usuario, Senha);
} catch (SQLException ex) {
    ex.printStackTrace();
}
```

3.3 – Execução de Procedimentos

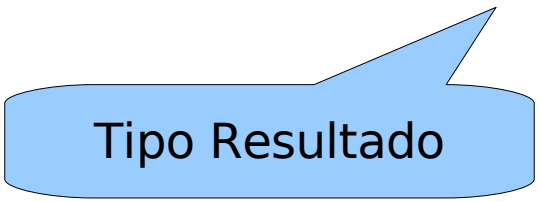
```
Statement st = null;
try {
    st = Conexao.createStatement();
    ...
    ...
    st.executeQuery(...) ---> ResultSet
    st.executeUpdate(...) --> (int) Linhas Afetadas

} catch (SQLException ex) {
    ex.printStackTrace();
}
```

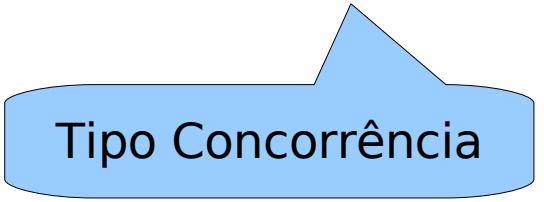


3.3.1 – ResultSet (Conjunto de Resultados)

- Possui melhor ou pior desempenho/mais ou menos recursos de consulta de acordo com as características do **statement;**

```
st = Conexao.createStatement([<Param1>,<Param2>]);
```






```
ResultSet rs = null;
rs = st.executeQuery(...);
```

3.3.1.1 Tipos de Resultados (Parâmetros)

<Param1>:

ResultSet.TYPE_FORWARD_ONLY --> somente se move para frente (JDBC 1.x);

ResultSet.TYPE_SCROLL_SENSITIVE --> atualiza o cache do ResultSet aberto;

ResultSet.TYPE_SCROLL_INSENSITIVE --> não atualiza o cache do ResultSet aberto;

<Param2>:

ResultSet.CONCUR_READ_ONLY --> as linhas do ResultSet são somente leitura;

ResultSet.CONCUR_UPDATABLE --> permite modificar;

4 - Métodos do ResultSet

- Ver API java;
- http://java.sun.com/j2se/1.5.0/docs/api/java/sql/ResultSet.html
-